

(11)特許出願公開番号

(43)公開日 平成6年(1994)3月18日

(51)Int.Cl. ⁵	識別記号	庁内整理番号	F I	技術表示箇所
H 0 3 H 17/00	B	7037-5 J		
G 1 1 B 5/09	3 2 1 A	8322-5D		
20/10	3 2 1 A	7923-5D		
20/18	1 0 2	9074-5D		
H 0 3 H 15/00		7037-5 J		

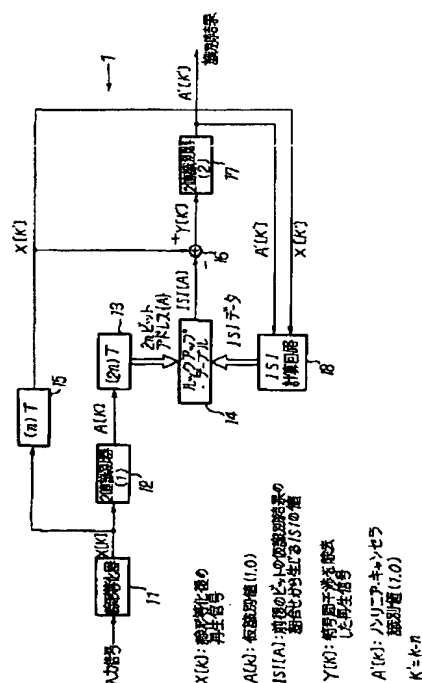
審査請求 未請求 請求項の数18(全 21 頁) 最終頁に続く

(21)出願番号	特願平4-250801	(71)出願人	000002185 ソニー株式会社 東京都品川区北品川6丁目7番35号
(22)出願日	平成4年(1992)8月26日	(72)発明者	原 雅明 東京都品川区北品川6丁目7番35号 ソニー株式会社内
		(74)代理人	弁理士 佐藤 隆久

(54)【発明の名称】 ノンリニアキャンセラー

(57) 【要約】

【構成】 線形等化器 11 と、その出力信号を仮識別する第一の 2 値識別器 12 と、前後数ビットの仮識別の結果の組み合わせによって生じる I S I のデータを記憶するルックアップテーブル 14 と、前後数ビットの仮識別結果の組み合わせをルックアップテーブル 14 の記憶手段のアドレスに変換し、その内容を読み出す手段と、ルックアップテーブル 14 から得られる符号間干渉 (I S I) のデータを線型等化器 11 の出力から減算する演算回路 16 と、演算回路 16 の出力を識別する第二の 2 値識別器 17 により再生信号から記録データを識別し、線型等化器 11 の出力信号と第二の 2 値識別器 17 の出力信号から符号間干渉のデータを計算し、ルックアップテーブル 14 の RAM の該当アドレスの内容を逐次更新する I S I 計算回路 18 により、自動的に I S I のデータを生成する。



【特許請求の範囲】

【請求項1】線形等化器と、この線形等化器の出力信号を仮識別する識別手段と、前後数ビットの前記仮識別の結果の組み合わせによって生じる符号間干渉のデータを記憶する、記憶手段により構成されるルックアップテーブルと、前後数ビットの仮識別結果の組み合わせを前記ルックアップテーブルの記憶手段のアドレスに変換し、前記ルックアップテーブルの内容を読み出す手段と、前記ルックアップテーブルから得られる符号間干渉のデータを前記線形等化器の出力から減算する手段と、この減算する手段の出力を識別する手段と、前記線形等化器の出力信号と前記識別する手段の出力信号から符号間干渉のデータを計算し、ルックアップテーブルの記憶手段の該当アドレスの内容を逐次更新する符号間干渉のデータの計算手段とを有するノンリニアキャンセラー。

【請求項2】請求項1記載のノンリニアキャンセラーにおいて、前記ルックアップテーブルの記憶手段に記憶される符号間干渉のデータの初期値は0であり、前記符号間干渉信号のデータの計算手段は前記線形等化器の出力信号と前記等化器の出力信号の仮識別の結果から、ある時点での符号間干渉のデータの値を計算し、この符号間干渉のデータを前記ルックアップテーブルの記憶手段の該当アドレスの仮の符号間干渉のデータの仮の値に順次加算し、この加算の回数が一定回数に達した場合に該当アドレスの仮の符号間干渉のデータをこの加算回数で除算し、この除算結果の値と前記ルックアップテーブルの記憶手段に記憶される符号間干渉のデータの平均値を新たな符号間干渉のデータとして、前記ルックアップテーブルの記憶手段の内容を更新することを特徴とするノンリニアキャンセラー。

【請求項3】請求項1記載のノンリニアキャンセラーにおいて、前記ルックアップテーブルの記憶手段に記憶される符号間干渉のデータの初期値は0であり、前記符号間干渉信号のデータの計算手段は前記線形等化器の出力信号と前記減算する手段の出力を識別する手段の出力信号から、ある時点での符号間干渉のデータの値を計算し、この符号間干渉のデータを前記ルックアップテーブルの記憶手段の該当アドレスの仮の符号間干渉のデータの仮の値に順次加算し、この加算の回数が一定回数に達した場合に該当アドレスの仮の符号間干渉のデータをこの加算回数で除算し、この除算結果の値と前記ルックアップテーブルの記憶手段に記憶される符号間干渉のデータの平均値を新たな符号間干渉のデータとして、前記ルックアップテーブルの記憶手段の内容を更新することを特徴とするノンリニア

キャンセラー。

【請求項4】請求項1記載のノンリニアキャンセラーにおいて、前記ルックアップテーブルの記憶手段に記憶される符号間干渉のデータの初期値は0であり、前記符号間干渉信号のデータの計算手段は前記線形等化器の出力信号と前記等化器の出力信号の仮識別の結果から、ある時点での符号間干渉のデータの値を計算し、この符号間干渉のデータを前記ルックアップテーブルの記憶手段の該当アドレスの仮の符号間干渉のデータの仮の値に順次加算し、加算開始から一定時間経過した場合に該当アドレスの仮の符号間干渉のデータをこの加算回数で除算し、この除算結果の値と前記ルックアップテーブルの記憶手段に記憶される符号間干渉のデータの平均値を新たな符号間干渉のデータとして、前記ルックアップテーブルの記憶手段の内容を更新することを特徴とするノンリニアキャンセラー。

【請求項5】請求項1記載のノンリニアキャンセラーにおいて、前記ルックアップテーブルの記憶手段に記憶される符号間干渉のデータの初期値は0であり、前記符号間干渉信号のデータの計算手段は前記線形等化器の出力信号と前記減算する手段の出力を識別する手段の出力信号から、ある時点での符号間干渉のデータの値を計算し、この符号間干渉のデータを前記ルックアップテーブルの記憶手段の該当アドレスの仮の符号間干渉のデータの仮の値に順次加算し、加算開始から一定時間経過した場合に該当アドレスの仮の符号間干渉のデータをこの加算回数で除算し、この除算結果の値と前記ルックアップテーブルの記憶手段に記憶される符号間干渉のデータの平均値を新たな符号間干渉のデータとして、前記ルックアップテーブルの記憶手段の内容を更新することを特徴とするノンリニアキャンセラー。

【請求項6】請求項1、請求項2、請求項3、請求項4、または請求項5記載のノンリニアキャンセラーにおいて、前記線形等化器の出力信号を仮識別する識別手段と前記減算する手段の出力を識別する手段は3値識別器であり、それぞれの前記3値識別器の出力側にパースナルレスポンス クラス1Vデコーダを配設したことを特徴とするノンリニアキャンセラー。

【請求項7】請求項2、請求項3、請求項4、請求項5、または請求項6記載のノンリニアキャンセラーにおいて、前記ルックアップテーブルの記憶手段に記憶される符号間干渉のデータの初期値はルックアップテーブルの更新前に記憶された符号間干渉データであることを特徴とす

るノンリニアキャンセラー

【請求項8】線形等化器の出力信号を仮識別し、識別結果とともに状態を出力するパーシャルレスポンス クラスIV用のビタビ復号器と、前記識別結果と前記状態の組み合わせをルックアップテーブルの記憶手段のアドレスに変換する手段と、前記識別結果を遅延する手段と、前記ルックアップテーブルの記憶手段に記憶された符号間干渉のデータを前記識別結果から減算する手段と、前記減算する手段の出力信号を識別するパーシャルレスポンス クラスIV用のビタビ復号器とを有することを特徴とするノンリニアキャンセラー。

【請求項9】請求項8記載のノンリニアキャンセラーにおいて、前記パーシャルレスポンス クラスIV用のビタビ復号器は、2並列のNRZ I用のビタビ復号器から構成されていることを特徴とするノンリニアキャンセラー。

【請求項10】識別結果と前記状態の組み合わせをルックアップテーブルの記憶手段のアドレスに変換する手段と、前記識別結果を遅延する手段と、前記ルックアップテーブルの記憶手段に記憶された符号間干渉のデータを前記識別結果から減算する手段と、前記減算する手段の出力信号を識別するパーシャルレスポンス クラスIV用のビタビ復号器とを一構成単位とし、この構成単位を多段接続し、前記多段接続された構成単位の入力側に線形等化器の出力信号を仮識別し、識別結果とともに状態を出力するパーシャルレスポンス クラスIV用のビタビ復号器を有することを特徴とするノンリニアキャンセラー。

【請求項11】請求項10記載のノンリニアキャンセラーにおいて、前記パーシャルレスポンス クラスIV用のビタビ復号器は、2並列のNRZ I用のビタビ復号器から構成されていることを特徴とするノンリニアキャンセラー。

【請求項12】請求項8、請求項9、請求項10、または請求項11記載のノンリニアキャンセラーにおいて、前記線形等化器の出力信号と前記識別するビタビ復号器の出力信号から符号間干渉のデータを計算し、ルックアップテーブルの記憶手段の該当アドレスの内容を逐次更新する符号間干渉のデータの計算手段とを有することを特徴とするノンリニアキャンセラー。

【請求項13】請求項12記載のノンリニアキャンセラーにおいて、前記ルックアップテーブルの記憶手段に記憶される符号間干渉のデータの初期値は0であり、前記符号間干渉信号のデータの計算手段は前記線形等化器の出力信号と、前記等化器の出力信号の仮識別の結果から、ある時点での符号間干渉のデータの値を計算し、この符号間干渉のデータを前記ルックアップテーブルの

記憶手段の該当アドレスの仮の符号間干渉のデータの仮の値に順次加算し、この加算の回数が一定回数に達した場合に、該当アドレスの仮の符号間干渉のデータをこの加算回数で除算し、

この除算結果の値と前記ルックアップテーブルの記憶手段に記憶される符号間干渉のデータの平均値を新たな符号間干渉のデータとして、前記ルックアップテーブルの記憶手段の内容を更新することを特徴とするノンリニアキャンセラー。

【請求項14】請求項12記載のノンリニアキャンセラーにおいて、

前記ルックアップテーブルの記憶手段に記憶される符号間干渉のデータの初期値は0であり、

前記符号間干渉信号のデータの計算手段は前記線形等化器の出力信号と、前記減算する手段の出力を識別する手段の出力信号から、ある時点での符号間干渉のデータの値を計算し、この符号間干渉のデータを前記ルックアップテーブルの記憶手段の該当アドレスの仮の符号間干渉のデータの仮の値に順次加算し、この加算の回数が一定回数に達した場合に、該当アドレスの仮の符号間干渉のデータをこの加算回数で除算し、

この除算結果の値と前記ルックアップテーブルの記憶手段に記憶される符号間干渉のデータの平均値を新たな符号間干渉のデータとして、前記ルックアップテーブルの記憶手段の内容を更新することを特徴とするノンリニアキャンセラー。

【請求項15】請求項12記載のノンリニアキャンセラーにおいて、

前記ルックアップテーブルの記憶手段に記憶される符号間干渉のデータの初期値は0であり、

前記符号間干渉信号のデータの計算手段は前記線形等化器の出力信号と、前記減算する手段の出力を識別する手段の出力信号から、ある時点での符号間干渉のデータの値を計算し、この符号間干渉のデータを前記ルックアップテーブルの記憶手段の該当アドレスの仮の符号間干渉のデータの仮の値に順次加算し、この加算の回数が一定回数に達した場合に、該当アドレスの仮の符号間干渉のデータをこの加算回数で除算し、

この除算結果の値と前記ルックアップテーブルの記憶手段に記憶される符号間干渉のデータの平均値を新たな符号間干渉のデータとして、前記ルックアップテーブルの記憶手段の内容を更新することを特徴とするノンリニアキャンセラー。

【請求項16】請求項12記載のノンリニアキャンセラーにおいて、

前記ルックアップテーブルの記憶手段に記憶される符号間干渉のデータの初期値は0であり、

前記符号間干渉信号のデータの計算手段は前記線形等化器の出力信号と、前記等化器の出力信号の仮識別の結果から、ある時点での符号間干渉のデータの値を計算し、

この符号間干渉のデータを前記ルックアップテーブルの記憶手段の該当アドレスの仮の符号間干渉のデータの仮の値に順次加算し、加算開始から一定時間経過した場合に、該当アドレスの仮の符号間干渉のデータをこの加算回数で除算し、

この除算結果の値と前記ルックアップテーブルの記憶手段に記憶される符号間干渉のデータの平均値を新たな符号間干渉のデータとして、前記ルックアップテーブルの記憶手段の内容を更新することを特徴とするノンリニアキャンセラー。

【請求項17】請求項12記載のノンリニアキャンセラーにおいて、

前記ルックアップテーブルの記憶手段に記憶される符号間干渉のデータの初期値は0であり、

前記符号間干渉信号のデータの計算手段は前記線形等化器の出力信号と、前記減算する手段の出力を識別する手段の出力信号から、ある時点での符号間干渉のデータの値を計算し、この符号間干渉のデータを前記ルックアップテーブルの記憶手段の該当アドレスの仮の符号間干渉のデータの仮の値に順次加算し、加算開始から一定時間経過した場合に、該当アドレスの仮の符号間干渉のデータをこの加算回数で除算し、

この除算結果の値と前記ルックアップテーブルの記憶手段に記憶される符号間干渉のデータの平均値を新たな符号間干渉のデータとして、前記ルックアップテーブルの記憶手段の内容を更新することを特徴とするノンリニアキャンセラー。

【請求項18】請求項14、請求項15、請求項16、または請求項17記載のノンリニアキャンセラーにおいて、

前記ルックアップテーブルの記憶手段に記憶される符号間干渉のデータの初期値はルックアップテーブルの更新前に記憶された符号間干渉データであることを特徴とするノンリニアキャンセラー

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、ノンリニアキャンセラー、特に磁気記憶装置の再生信号波形を等化し、記録データの識別を行うノンリニアキャンセラーに関する。

【0002】

【従来の技術】以下、従来の再生等化器について説明する。デジタルVTR等のデジタル磁気記録再生装置においては、デジタル形式の記録データは一度アナログ形式の信号に変換されて磁気記憶媒体上に記憶される。よって、上記デジタル形式の記録データの再生時には、時記憶媒体に記録される以前のデジタル形式の情報を磁気記憶媒体から検出されるアナログ形式の再生信号から得ることになる。このため、再生信号の波形をできるだけ符号間干渉（ISI）が少ないように整形する再生等化器と呼ばれるフィルターが用いられる。

【0003】この再生等化器には、通常、コイル（L）やコンデンサ（C）によるアナログフィルターやディレイラインを直列に結合して構成されるトランスバーサルフィルター等の線形等化器（LE）が用いられる。これらの線形等化器を通った後の再生信号を閾値識別器に入力し、ある閾値との大小から再生信号をデジタル形式のデータ（再生データ）に戻すことになる。

【0004】しかし、再生等化器として線形等化器を使用した場合、線形等化器が再生信号の信号成分とノイズ成分を区別せずに所望の周波数特性に合わせようとするので、ノイズが強調されてしまうという不具合が生じる。このため、再生等化回路の周波数特性とS/Nの適切なトレードオフを取って、最終的なデジタル形式の再生データのエラーレートが最小になるように調整する必要があるという問題点がある。この問題点を解決するためには、以下に述べる二つの方法がある。

【0005】以下、再生等化器としてビタビ復号器を用いる方法を説明する。1つめの方法は、ビタビ復号器を利用して再生信号の持つS/Nを最大限に利用して識別する方法である。ビタビ復号器では、あらかじめ規定されたISIの値によって区別されるn個の状態をそれ以前のmビットの識別値の組合せによって定義し、1ビット分の処理が終わるたびに前記のn個の状態は、次のn個の状態に更新される。前記n個の各状態は、これまでの識別値の履歴および尤度を持っている。

【0006】ノイズがガウス分布すると仮定すると、n個の各状態の尤度はノイズが存在しないときの再生信号の値と実際の再生信号の値との差の2乗のこれまでの和になる。前記n個の各状態は、可能性のあるすべての前の状態の中で最も尤度が大きくなるものから推移したものと判断して、前の状態から次の状態に更新されるとともに、識別値の履歴および尤度も更新される。

【0007】このように最も尤もらしい状態推移を繰り返していくと、ある段階で数ビット前までの履歴が一つの履歴に統一され、それまでの識別値が確定する。この方法では再生信号の信号電力を最大限有効に利用して識別するので、通常の閾値識別に比べて非常に良好なエラーレートが得られる。

【0008】しかし、2乗和を計算する必要があるために回路規模が大きくなり、なおかつ、データレートのクロックで動作させることが困難であるという大きな問題点があるので、多状態のビタビ復号器はデジタルVTRなどでは実用化されていない。

【0009】ビタビ復号器の最も簡単な例として、NRZI方式への適用がある。NRZI方式で記録再生し、単位パルスが（1，-1）の値をとるように等化すると状態数は2状態になり、2乗和を計算する必要も無くなるので簡単にビタビ復号器を構成することができる。さらに、PRIV方式を用いて、記録データを2ビット遅延とmod 2加算によるプリコーディングしてから記録再

生し、単位パルスが(1, 0, -1)の値をとるように等化すると、1ビットおきに見ればNRZになる。

【0010】したがって、NRZ用の簡単なビタビ復号器を2個並列に用い、データレートの半分の速さで動作させればNRZ方式への適用が可能となる。このPRIV方式とビタビ復号器の組合せは、回路規模および動作速度ともに実用的なビタビ復号器を構成することができるので、最近のデジタルVTRにおいて一般的になりつつある。ここまでの説明した技術の内容については一般に知られており、例えば日刊工業新聞社刊、「デジタルビデオ記録技術」等に記載されている。

【0011】以下、PRIV用のビタビ復号器のことをVDとも記す。以上述べたVDでは、ノイズに相関がなく、ガウス分布すると仮定すると、理論的には閾値識別に比べて3dBのS/N改善効果がある。しかし、線形等化器を通すことによってノイズは相関を持ち、また、実際にPRIVの基準通りに等化することは困難なので、ビタビ復号器で期待されるだけの改善効果を実現できないという問題点があった。

【0012】以下、再生等化器として非線形等化器を用いる方法を説明する。もう一つの方法は、リニアキャンセラー(LC)やノンリニアキャンセラー(NLC)等の非線形等化器を用いて、ノイズを強調することなく符号間干渉(ISI)を抑圧する方法である。これらは前後のビットの仮識別値の組合せによって決定するISIのコピーをあらかじめ設定しておいて、これを線形等化器の出力信号から差し引いてからもう一度識別するものである。LCが線形歪みに対してのみ有効であるのに対し、NLCは非線形歪みに対しても有効な方式である。

【0013】NLCを使用した再生等化器の構成方法については種々の方法があるが、等化誤差によって生じるISIの長さを十分に考慮でき、なおかつ回路構成が簡単になるものとして、テーブルルックアップ型NLCがある。テーブルルックアップ型NLCは、あらかじめISIのデータを格納したRAMをルックアップテーブルとし、線形等化器の出力信号にもとづいて閾値識別器によって識別した前後のビットの仮識別値の組合せをアドレスに変換してISIの値を読み出し、これを再生信号から差し引いたのち、もう一度閾値識別器で識別するものである。

【0014】図12は従来の再生等化器7の構成を示す図である。以下、図12を参照して従来の再生等化器7の動作を説明する。線形等化器11の再生信号X[k]にもとづいて第一の2値識別器12で仮識別された仮識別値A[k] ($A[k] = 1 \text{ or } 0$) は、2n段のディレーライン13によりルックアップテーブル14のアドレス{A}になる。

【0015】アドレス{A}は2nビットであり、ISIを差し引く前の再生信号X[k]を識別した結果である(2n+1)個の仮識別値A[i] ($i = k, k -$

$1, \dots, k - 2n$) から仮識別値A[k'] (ただし、 $k' = k - n$)を除いた2n個の仮識別値A[i]によって決定される。

【0016】前後nビットずつの仮識別値の組合せによって生じるISIを除去するために、ルックアップテーブル14に内蔵されるRAM(図示せず)は2nのISIデータを記憶できる容量を持つ必要がある。ルックアップテーブル14から読みだされたアドレス{A}のISIデータISI{A}を、演算回路16においてn段のディレーライン15の出力X[k']から差し引いてISIが除去された再生信号Y[k']を作り、これを第二の2値識別器17で識別した識別値A'[k']を最終的な識別値とする。

【0017】この従来の再生等化器7では、ルックアップテーブル14用のRAMを持つだけなので回路構成は非常に簡単であり、十分に長い前後のビットを考慮してISIを除去するようなNLCを容易に実現することができる。

【0018】しかし、あらかじめISIのデータをRAMに書き込んでおく必要があり、調整すべきISIのデータは2の2n乗で増えていくので、例えば、前後5ビットずつを考慮すると1024通りのISIデータを設定する必要がある。

【0019】このISIデータを一つ一つ調整していくには大きな労力が生じる。よって、実用上前後2ビットずつの16通り程度が限界であり、従来の再生等化器7の回路構成上のメリットを十分に活かさない。この問題点を解決するためには、ISIのデータを自動的に設定する方法があればよい。しかし、これまでISIのデータを自動的に設定する方法はなかった。

【0020】また、非線形等化器に含まれるLCおよびNLCのいずれにおいても、適切なISIの値を再生信号から差し引くためには仮識別値に含まれるエラーが少ないことが前提条件となる。しかし、仮識別値にエラーは必然的に含まれているので、実際の条件下においてはNLCによる改善効果が劣化し、またエラーが伝播するという問題点があった。

【0021】

【発明が解決しようとする課題】以上述べたように、従来の再生等化方法は、再生等化器として線型等化器(LC)、ビタビ復号器(VD)、あるいは非線形等化器(NLC)を使用している。このため、以上に述べたように、再生等化回路として線型等化器を使用した場合、線形等化器が再生信号の信号成分とノイズ成分を区別せずに所望の周波数特性に合わせようとするので、ノイズが強調されてしまうという不具合が生じる。このため、再生等化回路の周波数特性とS/Nの適切なトレードオフを取って、最終的なデジタル形式の再生データのエラーレートが最小になるように調整する必要があるという問題点があった。

【0022】また、以上に述べたように、再生等化器としてVDを用いた再生等化方法においては、線形等化器を通すことによってノイズは相関を持ち、また、実際にPRIVの基準通りに等化することは困難なので、VDで期待されるだけの改善効果を実現できないという問題点があった。

【0023】また、以上述べたように、再生等化器として非線形復号器を用いた再生等化方法においては、あらかじめISIのデータをRAMに書き込んでおく必要があり、調整すべきISIのデータが2の2n乗で増えていくので、例えば、前後5ビットずつを考慮すると1024通りのISIデータを設定する必要がある。

【0024】このISIデータを一つ一つ調整していくには大きな労力が生じる。よって、実用上前後2ビットずつの16通り程度が限界であり、回路構成上のメリットを十分に生かせないという問題点があった。

【0025】また、LCおよびNLCのいずれにおいても、適切なISIの値を再生信号から差し引くためには仮識別値に含まれるエラーが少ないことが前提条件となる。しかし、仮識別値にエラーは必然的に含まれているので、実際の条件下においてはNLCによる改善効果が劣化し、またエラーが伝播するという問題点があった。

【0026】本発明は、以上述べたような従来の技術の問題点に鑑みてなされたものであり、テーブルルックアップ型のNLCにおいて、初期設定も含めて外部からの調整を必要とせず、また、十分に長い前後のビットを考慮でき、また、テープ、ヘッドの特性の変化やばらつきを補償することができ、また、仮識別値に含まれるエラーを少なくすることができ、また、仮識別値にエラーが含まれていてもエラーの伝播が少なく、また、符号間干渉が減少するとともにノイズの相関が減少し、信号電力有効利用を図ることができ、また、簡単な回路構成で実現でき、良好なエラーレートが得られるノンリニアキャンセラーを提供することを目的とする。

【0027】

【課題を解決するための手段】以上に述べた課題を解決するため、本発明のノンリニアキャンセラーは、線形等化器と、この線形等化器の出力信号を仮識別する識別手段と、前後数ビットの前記仮識別の結果の組み合わせによって生じる符号間干渉のデータを記憶する、記憶手段により構成されるルックアップテーブルと、前後数ビットの仮識別結果の組み合わせを前記ルックアップテーブルの記憶手段のアドレスに変換し、前記ルックアップテーブルの内容を読み出す手段と、前記ルックアップテーブルから得られる符号間干渉のデータを前記線形等化器の出力から減算する手段と、この減算する手段の出力を識別する手段と、前記線形等化器の出力信号と前記識別する手段の出力信号から符号間干渉のデータを計算し、ルックアップテーブルの記憶手段の該当アドレスの内容を逐次更新する符号間干渉のデータの計算手段とを有す

る。

【0028】また、前記ルックアップテーブルの記憶手段に記憶される符号間干渉のデータの初期値は0であり、前記符号間干渉信号のデータの計算手段は前記線形等化器の出力信号と前記等化器の出力信号の仮識別の結果から、ある時点での符号間干渉のデータの値を計算し、この符号間干渉のデータを前記ルックアップテーブルの記憶手段の該当アドレスの仮の符号間干渉のデータの仮の値に順次加算し、この加算の回数が一定回数に達した場合に該当アドレスの仮の符号間干渉のデータをこの加算回数で除算し、この除算結果の値と前記ルックアップテーブルの記憶手段に記憶される符号間干渉のデータの平均値を新たな符号間干渉のデータとして、前記ルックアップテーブルの記憶手段の内容を更新することを特徴とする。

【0029】また、前記ルックアップテーブルの記憶手段に記憶される符号間干渉のデータの初期値は0であり、前記符号間干渉信号のデータの計算手段は前記線形等化器の出力信号と前記減算する手段の出力を識別する手段の出力信号から、ある時点での符号間干渉のデータの値を計算し、この符号間干渉のデータを前記ルックアップテーブルの記憶手段の該当アドレスの仮の符号間干渉のデータの仮の値に順次加算し、この加算の回数が一定回数に達した場合に該当アドレスの仮の符号間干渉のデータをこの加算回数で除算し、この除算結果の値と前記ルックアップテーブルの記憶手段に記憶される符号間干渉のデータの平均値を新たな符号間干渉のデータとして、前記ルックアップテーブルの記憶手段の内容を更新することを特徴とする。

【0030】また、前記ルックアップテーブルの記憶手段に記憶される符号間干渉のデータの初期値は0であり、前記符号間干渉信号のデータの計算手段は前記線形等化器の出力信号と前記等化器の出力信号の仮識別の結果から、ある時点での符号間干渉のデータの値を計算し、この符号間干渉のデータを前記ルックアップテーブルの記憶手段の該当アドレスの仮の符号間干渉のデータの仮の値に順次加算し、加算開始から一定時間経過した場合に該当アドレスの仮の符号間干渉のデータをこの加算回数で除算し、この除算結果の値と前記ルックアップテーブルの記憶手段に記憶される符号間干渉のデータの平均値を新たな符号間干渉のデータとして、前記ルックアップテーブルの記憶手段の内容を更新することを特徴とする。

【0031】また、前記ルックアップテーブルの記憶手段に記憶される符号間干渉のデータの初期値は0であり、前記符号間干渉信号のデータの計算手段は前記線形等化器の出力信号と前記減算する手段の出力を識別する手段の出力信号から、ある時点での符号間干渉のデータの値を計算し、この符号間干渉のデータを前記ルックアップテーブルの記憶手段の該当アドレスの仮の符号間干

渉のデータの仮の値に順次加算し、加算開始から一定時間経過した場合に該当アドレスの仮の符号間干渉のデータをこの加算回数で除算し、この除算結果の値と前記ルックアップテーブルの記憶手段に記憶される符号間干渉のデータの平均値を新たな符号間干渉のデータとして、前記ルックアップテーブルの記憶手段の内容を更新することを特徴とする。

【0032】また、前記線形等化器の出力信号を仮識別する識別手段と前記減算する手段の出力を識別する手段は3値識別器であり、それぞれの前記3値識別器の出力側にパシシャルレスポンス クラスIVデコーダを配設したことを特徴とする。

【0033】また、前記ルックアップテーブルの記憶手段に記憶される符号間干渉のデータの初期値はルックアップテーブルの更新前に記憶された符号間干渉データであることを特徴とする。

【0034】また、線形等化器の出力信号を仮識別し、識別結果とともに状態を出力するパシシャルレスポンス クラスIV用のビタビ復号器と、前記識別結果と前記状態の組み合わせをルックアップテーブルの記憶手段のアドレスに変換する手段と、前記識別結果を遅延する手段と、前記ルックアップテーブルの記憶手段に記憶された符号間干渉のデータを前記識別結果から減算する手段と、前記減算する手段の出力信号を識別するパシシャルレスポンス クラスIV用のビタビ復号器とを有することを特徴とする。

【0035】また、前記パシシャルレスポンス クラスIV用のビタビ復号器は、2並列のNRZI用のビタビ復号器から構成されていることを特徴とする。

【0036】また、識別結果と前記状態の組み合わせをルックアップテーブルの記憶手段のアドレスに変換する手段と、前記識別結果を遅延する手段と、前記ルックアップテーブルの記憶手段に記憶された符号間干渉のデータを前記識別結果から減算する手段と、前記減算する手段の出力信号を識別するパシシャルレスポンス クラスIV用のビタビ復号器とを一構成単位とし、この構成単位を多段接続し、前記多段接続された構成単位の入力側に線形等化器の出力信号を仮識別し、識別結果とともに状態を出力するパシシャルレスポンス クラスIV用のビタビ復号器を有することを特徴とする。

【0037】また、前記パシシャルレスポンス クラスIV用のビタビ復号器は、2並列のNRZI用のビタビ復号器から構成されていることを特徴とする。

【0038】また、前記線形等化器の出力信号と前記識別するビタビ復号器の出力信号から符号間干渉のデータを計算し、ルックアップテーブルの記憶手段の該当アドレスの内容を逐次更新する符号間干渉のデータの計算手段とを有することを特徴とする。

【0039】また、前記ルックアップテーブルの記憶手段に記憶される符号間干渉のデータの初期値は0であ

り、前記符号間干渉信号のデータの計算手段は前記線形等化器の出力信号と、前記等化器の出力信号の仮識別の結果から、ある時点での符号間干渉のデータの値を計算し、この符号間干渉のデータを前記ルックアップテーブルの記憶手段の該当アドレスの仮の符号間干渉のデータの仮の値に順次加算し、この加算の回数が一定回数に達した場合に、該当アドレスの仮の符号間干渉のデータをこの加算回数で除算し、この除算結果の値と前記ルックアップテーブルの記憶手段に記憶される符号間干渉のデータの平均値を新たな符号間干渉のデータとして、前記ルックアップテーブルの記憶手段の内容を更新することを特徴とする。

【0040】また、前記ルックアップテーブルの記憶手段に記憶される符号間干渉のデータの初期値は0であり、前記符号間干渉信号のデータの計算手段は前記線形等化器の出力信号と、前記減算する手段の出力を識別する手段の出力信号から、ある時点での符号間干渉のデータの値を計算し、この符号間干渉のデータを前記ルックアップテーブルの記憶手段の該当アドレスの仮の符号間干渉のデータの仮の値に順次加算し、この加算の回数が一定回数に達した場合に、該当アドレスの仮の符号間干渉のデータをこの加算回数で除算し、この除算結果の値と前記ルックアップテーブルの記憶手段に記憶される符号間干渉のデータの平均値を新たな符号間干渉のデータとして、前記ルックアップテーブルの記憶手段の内容を更新することを特徴とする。

【0041】また、前記ルックアップテーブルの記憶手段に記憶される符号間干渉のデータの初期値は0であり、前記符号間干渉信号のデータの計算手段は前記線形等化器の出力信号と、前記減算する手段の出力を識別する手段の出力信号から、ある時点での符号間干渉のデータの値を計算し、この符号間干渉のデータを前記ルックアップテーブルの記憶手段の該当アドレスの仮の符号間干渉のデータの仮の値に順次加算し、この加算の回数が一定回数に達した場合に、該当アドレスの仮の符号間干渉のデータをこの加算回数で除算し、この除算結果の値と前記ルックアップテーブルの記憶手段に記憶される符号間干渉のデータの平均値を新たな符号間干渉のデータとして、前記ルックアップテーブルの記憶手段の内容を更新することを特徴とする。

【0042】また、前記ルックアップテーブルの記憶手段に記憶される符号間干渉のデータの初期値は0であり、前記符号間干渉信号のデータの計算手段は前記線形等化器の出力信号と、前記等化器の出力信号の仮識別の結果から、ある時点での符号間干渉のデータの値を計算し、この符号間干渉のデータを前記ルックアップテーブルの記憶手段の該当アドレスの仮の符号間干渉のデータの仮の値に順次加算し、加算開始から一定時間経過した場合に、該当アドレスの仮の符号間干渉のデータをこの加算回数で除算し、この除算結果の値と前記ルックアッ

プテーブルの記憶手段に記憶される符号間干渉のデータの平均値を新たな符号間干渉のデータとして、前記ルックアップテーブルの記憶手段の内容を更新することを特徴とする。

【0043】また、前記ルックアップテーブルの記憶手段に記憶される符号間干渉のデータの初期値は0であり、前記符号間干渉信号のデータの計算手段は前記線形等化器の出力信号と、前記減算する手段の出力を識別する手段の出力信号から、ある時点での符号間干渉のデータの値を計算し、この符号間干渉のデータを前記ルックアップテーブルの記憶手段の該当アドレスの仮の符号間干渉のデータの仮の値に順次加算し、加算開始から一定時間経過した場合に、該当アドレスの仮の符号間干渉のデータをこの加算回数で除算し、この除算結果の値と前記ルックアップテーブルの記憶手段に記憶される符号間干渉のデータの平均値を新たな符号間干渉のデータとして、前記ルックアップテーブルの記憶手段の内容を更新することを特徴とする。

【0044】また、前記ルックアップテーブルの記憶手段に記憶される符号間干渉のデータの初期値はルックアップテーブルの更新前に記憶された符号間干渉データであることを特徴とする。

【0045】

【作用】線形等化器の出力信号の値と識別値からISIを計算し、RAMの該当するアドレスのISIを逐次更新するISI計算回路により、初期設定も含めて外部からの調整を一切必要とせず、必要なISIの値を自動的に得ることができる。

【0046】また、PRIV用のビタビ復号器(VD)とNLCを組み合わせることにより、再生等化器の回路構成を簡単にし、良好なエラーレートを得る。また、NLCの仮識別器としてVDを用い、その識別値をもとにNLCでISIを除去した後もう一度VDで識別することにより、さらに良好なエラーレートを得る。

【0047】

【実施例】以下、本発明の第一の実施例について説明する。図1は本発明の第一の再生等化器1の構成を示す図である。第一の再生等化器1は図12の従来の再生等化器7に、n段の第二のディレーライン15を通った後の線形等化器11の出力X[k']とNLCの識別値A'[k']からISIデータを計算してルックアップテーブルのRAMに書き込む機能を持ったISI計算回路18を付加した構成になっている。

【0048】図1において、線型等化器11は、アナログフィルタ等で構成され、再生信号の波形成形を行う線型等化回路である。第一の2値識別器12は、線型等化器11の出力を仮識別する2値識別器である。2n段のディレーライン13は、第一の2値識別器12から出力される仮識別値からルックアップテーブル14用のアドレスを生成する遅延回路である。

【0049】ルックアップテーブル14は、RAMおよびその周辺回路から構成され、ISIの値を記憶する記憶回路である。n段のディレーライン15は、線型等化器11の出力を遅延させ、演算回路16に入力する遅延回路である。演算回路16は、n段のディレーライン15の出力からルックアップテーブル14の出力を減算する演算回路である。第二の2値識別器17は、演算回路16の出力を識別し、最終的な識別値を出力する2値識別器である。以上述べた第一の再生等化器1の各部分は、従来の再生等化器7の同一符号を付した部分と同じである。ISI計算回路18は、ISIデータを計算してルックアップテーブルのRAMに書き込む機能を持った計算回路である。

【0050】以下、X[]で示される信号は線形等化後の再生信号、A[]で示される信号は仮識別値、ISI[]で示される信号は前後のビットの組み合わせから生じる符号間干渉(ISI)の値、Y[]で示される信号は符号間干渉を除去した再生信号、A'[]で示される信号はノンリニアキャンセラ識別値である。また、(k' = k - n)である。

【0051】図2は、ISI計算回路18の構成を示す図である。図2において、RAM180は、仮ISIの値LISI{A'}とアドレス{A'}の発生回数N{A'}を格納するRAMである。ROM181は、計算プログラムを記憶するROMである。CPU182は、実際の計算を行なうCPUである。

【0052】以下、第一の再生等化器1の動作について説明する。まず、ISI計算回路18における、ISIデータの計算方法を説明する。最初に、RAM180のすべてのISIの値LISI{A'}および、仮ISIの値LISI{A'}、アドレス{A'}の発生回数N{A'}を0に(クリア)する。つまり、

ISI{A'} ← 0

LISI{A'} ← 0

N{A'} ← 0

のように、すべてのISIデータISI{A'}、仮ISIデータLISI{A'}、およびアドレス{A'}の発生回数N{A'}のRAM180上の記憶領域を0にする。

【0053】ここで、識別値A'[k' - n]を除く2n個の識別値A'[j] {j = k', k' - 1, ..., k' - 2n}によって決定するアドレスをアドレス{A'}とし、仮識別値A[k']を除く2n個の仮識別値A[i] {i = k, k - 1, ..., k - 2n}によって決定するアドレス{A}と区別して用いる。

【0054】次に、仮識別信号X[k' - n]と識別値B'[k' - n]の差をある瞬間のISIの値として、該当するアドレスのLISI{A'}に加え、アドレス{A'}の発生回数N{A'}を1増やす。つまり、

$$LISI\{A'\} \leftarrow LISI\{A'\} + X[k' - n] - B'[k' - n]$$

$$N\{A'\} \leftarrow N\{A'\} + 1$$

のようにRAM180の記憶内容を変更する。

【0055】ここで、説明の便宜上線型等化器11の再生信号 $X[k]$ の振幅は、 ± 1 に規格化されており、0を閾値にして1と0に識別されている。このとき、識別値 $A'[k] = 1$ に識別値 $B'[k] = 1$ を、識別値 $A'[k] = 0$ に識別値 $B'[k] = -1$ を対応させると、識別値 $B'[k] = \pm 1$ が線型等化器11の再生信号 $X[k]$ の目標値になるので、 $(X[k] - B'$

$$LISI\{A'\} \leftarrow (LISI\{A'\} + LISI\{A'\} / M) / 2$$

$$N\{A'\} \leftarrow 0$$

のようにRAM180の記憶内容を変更する。

【0057】ここでは仮 $LISI$ の平均値とRAMの $LISI$ の値を同じ比率0.5:0.5で加えて平均を取っているが、この比率は $(x:1-x)$ であれば任意のものでよい。また、RAMの動作速度が遅く1クロックの間で $LISI$ データの読み出しと書き込みをできないような場合には、ヘッド(図示せず)とテープ(図示せず)が接触している期間は仮 $LISI$ の値 $LISI\{A'\}$ とアドレス $\{A'\}$ の発生回数 $N\{A'\}$ を更新するのみにし、ヘッドとテープが接触していない間に $LISI$ データを更新してRAMに書き込めばよい。

【0058】以上述べたような非常に簡単な計算で、必要な $LISI$ データを自動的に生成することができる。なお、最初はすべての $LISI$ データは0になっているので、線形等化器11による仮識別値 $A[k]$ と同じものが識別値 $A'[k]$ に出力される。したがって、何回か $LISI$ データの更新を行なうまでは第一の再生等化器1の効果を得ることはできない。

【0059】しかし、線形等化器11による仮識別値に多くの誤りが含まれているような場合、適切な $LISI$ を読み出して除去することはできないので、ある程度良好な調整がなされた線形等化器11を用いることは第一の再生等化器1にとって必要なことである。したがって、本発明によって新たに発生した欠点にはならない。むしろ、初期設定が不要であるメリットのほうが大きい。言うまでもないが、以前の $LISI$ データがあるときは、 $LISI\{A'\}$ の初期値にはそれを使えばよい。

【0060】 $LISI$ 計算回路18により以上に述べたような方法で得られた $LISI$ データは、第一の再生等化器1において、従来の再生等化器7で説明したのと同様な方法で、再生信号の識別に使用される。

【0061】以下、本発明の第二の実施例について説明する。第一の実施例において説明した $LISI$ 計算回路18による $LISI$ の生成方法を $LISI$ 生成の基本方法とする。基本方法においては、 $LISI$ の値を差し引いた後の演算回路16の再生信号 $Y[k']$ を識別した結果である識別値 $A'[k']$ を用いて、ある瞬間 k' の $LISI$

$[k']$ がある瞬間 k の $LISI$ の値になる。

【0056】次に、あるアドレスの仮 $LISI$ があらかじめ設定された回数 M だけ加えられた場合、仮 $LISI$ を各アドレスの加算回数で割って仮 $LISI$ の平均値を計算し、これとRAMの $LISI$ とを加えてその平均を取り、これを新たにRAMに書き込むとともに、アドレス $\{A'\}$ の発生回数 $N\{A'\}$ を0にする。つまり、

L を計算する。

【0062】さらに、識別値 $A'[k' - n]$ を除いた $2n$ 個の識別値 $A'[j]$ ($j = k', k' - 1, \dots, k' - 2n$)によって $2n$ ビットのアドレス $\{A'\}$ を決定し、 $LISI$ データを作成してルックアップテーブル14に書き込む。

【0063】仮識別値 $A[k']$ を除いた $2n$ 個の仮識別値 $A[i]$ ($i = k, k - 1, \dots, k - 2n$)によって指定される $2n$ ビットのアドレス $\{A\}$ の $LISI$ データを読み出して再生信号 $X[k']$ から差し引いている。以下に説明する第二の実施例における $LISI$ データの決定方法は、基本方法の変形例である。

【0064】識別値 $A'[k' - n]$ を含む $(2n + 1)$ 個の識別値 $A'[j]$ ($j = k', k' - 1, \dots, k' - 2n$)によって $(2n + 1)$ ビットのアドレス $\{A'\}$ を決定して $LISI$ データを作成してルックアップテーブル14に書き込む。

【0065】これを読み出すときには仮識別値 $A[k']$ を除く仮識別値 $A[i]$ ($i = k, k - 1, \dots, k - 2n$)で仮識別値 $A[k'] = 1$ のときの $LISI\{A+\}$ と仮識別値 $A[k'] = 0$ のときの $LISI\{A-\}$ の二つのデータを同時に読み出し、この二つを平均してから再生信号 $X[k']$ から差し引く。

【0066】この場合、 2^{2n} 個の $LISI$ データを格納するRAMを2つ用意して2並列にし、 $LISI$ データ作成の際に該当識別値 $A'[k']$ が1か0かに応じて使い分ける。この方法では回路規模は大きくなるが、 $LISI$ データの信頼性が向上する。

【0067】以下、本発明の第三の実施例について説明する。図3は、本発明の第二の再生等化器2の構成を示す図である。第二の再生等化器2の各部分は、第一の再生等化器1の同一符号を付した部分に同じである。第一の再生等化器1の構成では $LISI$ を差し引いた後の演算回路16の再生信号 $Y[k']$ を識別した識別値 $A'[k']$ を用いて $LISI$ データを作成してしている。これは線形等化器11の出力結果よりも第一の再生等化器1の識別値のほうが信頼性が高いからである。しかし、

テープに欠陥が多い場合、あるいは再生信号のエンベロープ変動が大きい場合には、突発的にNL Cの出力結果のほうが信頼性が低くなることがある。

【0068】このような場合、演算回路16の再生信号 $Y[k']$ を識別した結果からISIデータを作成すると、演算回路16の再生信号 $Y[k']$ が不適切な値を示し始めた場合にISIデータが収束しなくなってしまう危険がある。そこで、第二の再生等化器2においては、ISIデータの作成に線形等化器11の再生信号 $X[k]$ を識別した結果を用いている。

【0069】以上述べた第二の再生等化器2については、ISI計算回路18における計算方法について、基本方法と第二の実施例におけるISIデータの決定方法の両方が適用可能である。同様な考え方から、第二の再生等化器2の構成は第一の再生等化器1と同様とし、再生信号のエンベロープを常に監視し、このエンベロープが不良になった場合にはLISI $\{A'\}$ の計算をストップしてもよい。

【0070】また、後段のECC回路でエラーレートが悪くなってことを検知して、その間はLISI $\{A'\}$ の計算をストップしてもよい。また、LISI $\{A'\}$ の計算をストップする代わりに、第一の再生等化器1と第二の再生等化器2構成を切り替えるように構成してもよい。また、エラーレートが劣化し続ける場合には、いったんルックアップテーブル14のISIデータをすべてクリアする方法をとるのも有効である。

【0071】以下、第四の実施例について説明する。以上に述べた各実施例においては、2値識別器を用いて再生信号を(1, 0)の値に識別する場合に限って説明した。第四の実施例においては、第一の再生等化器1および第二の再生等化器2の応用例として、第一の再生等化器1および第二の再生等化器2のPRIV(パースナルレスポンス クラスIV)への適用を行う。

【0072】図4は、本発明の第三の再生等化器3の構成を示す図である。図4において、第一の3値識別器21は、線形等化器11の出力を仮識別する3値識別器である。第二の3値識別器22は、演算回路16の出力を識別する3値識別器である。3値識別器21、22は±0.5を閾値として(1, 0, -1)の値に識別するものである。第一のPRIVデコーダ22は、必要なISIデータを読み出すために仮識別値 $A[k]$ からアドレス $\{A\}$ を発生させるためのものである。

【0073】第一のPRIVデコーダ23は、第一の3値識別器21の出力をPRIVデコードするデコーダである。第二のPRIVデコーダ24は、第一の3値識別器22の出力をPRIVデコードするデコーダである。これによりISI計算回路18で必要になるアドレス

$\{A'\}$ を発生させるとともに、1と0からなる最終的な識別値 $A'[k]$ を得る。上記以外の第三の再生等化器3の各部分の構成は第一の再生等化器1および第二の

再生等化器2の同一符号を付した各部分に同じである。

【0074】以下、 $B[\]$ で示される信号は仮識別値、 $B'[\]$ で示される信号はノンリニアキャンセラ識別値、 $A'[\]$ はPRIVデコード後のノンリニアキャンセラ識別値である。

【0075】第三の再生等化器3は第一の再生等化器1と類似の構成となっている。第一の再生等化器1をPRIVに適用するためには、以上に述べたような仮識別値 $B[k]$ の組合せから $2n+2$ ビットのアドレス $\{A\}$ を発生するようなデコーダを付加する必要がある。つまり、第三の再生等化器3は2つの3値識別器21、22と2種類のPRIVデコーダ23、24がある点で、第一の再生等化器1と異なっている。なお、再生信号 $X[k]$ の目標値としては識別値 $B'[k]$ がそのまま用いられる。このPRIVへの応用例に関しても、上記基本方法および第二の実施例および第三の実施例で述べたISIデータの決定方法の組合せが可能である。また、PRIV以外に適用する場合も、同様なデコーダを用いて識別値を必要最小限なビット数のアドレスに変換すればよい。

【0076】PRIVでは、記録データを2ビット遅延(mod 2)加算によるプリコーディングをしてから記録する。単位パルスに対する再生信号の応答が(1, 0, -1)になるように線形等化器11を用いて等化した再生信号 $X[k]$ を、±0.5を閾値として1, 0, -1からなる仮識別値 $B[k]$ に識別したのち、1と-1を1に0を0にデコードして仮識別値 $A[k]$ とする。

【0077】第一の実施例または第二の実施例にISIデータ計算方法に基づいて、 $2n+1$ 個の仮識別値 $B[k]$ をそのまま用いてアドレスを決定すると、これは3値なので 3^{2n+1} の組合せがある。しかしPRIVの場合、奇数系列と偶数系列は独立したNRZIであると見なすことが可能である。よって、2値(1と0)にデコードした後の 2^{2n+1} の組合せに加えて、奇数系列と偶数系列それぞれのNRZIの状態(1または0の2状態ずつ)を考えればよい。

【0078】着目しているビットに最も近い奇数個前のビットの非0が(-1)であれば($A_{odd}=0$)とし、1であれば($A_{odd}=1$)とする。同様に着目しているビットに最も近い偶数個前のビットの非0が-1であれば、($A_{even}=0$)とし、1であれば($A_{even}=1$)とする。

【0079】以上に述べた方法により、 $(2n+1)$ 個の仮識別値によって発生する組合せは 2^{2n+3} 通りであり、仮識別値 $B[k']$ は改めて識別するから無視すると、 $2n+2$ ビットのアドレスになる。図5に $n=3$ の場合を例に、仮識別値 $B[k]$ と仮識別値 $A[k]$ および $(2n+2)$ ビットのアドレス $\{A\}$ の関係の例を示す。

【0080】実験用デジタルVTRに511周期のM系列を記録し、等化前の再生信号を高速でAD変換して計算機に取り込んで、3～15タップのトランスバースルフィルタでPRIVの基準に等化し、識別値を記録データと比較したときのエラーレートを図6に示す。

【0081】図6においてNLCと示したものが、第三の再生等化器3をPRIVに応用した場合の結果である。連続した再生信号データの数約20000個で、ISIデータの初期値はすべて0を入れてM=64で更新している。この中で最も大きなエラーレートの向上が得られているのは7タップのトランスバースルフィルタにn=3のノンリニアキャンセラーを適用した場合であり、エラーレートが 2.38×10^{-3} から 2.46×10^{-4} になり、約(1/10)にエラーが減少している。トランスバースルフィルタのタップ数が多くなるに従い等化誤差が小さくなるので効果も小さくなるが、n=3のノンリニアキャンセラーを用いることで約(1/4)にエラーが減少している。

【0082】以下、第五の実施例について説明する。図7は、本発明の第四の再生等化器4の構成を示す図である。図7において、第一のビタビ復号器(VD)25は、線形等化器11の出力信号X[k]に基づいて仮識別し、1, 0の2値の仮識別値B[k-m]と各瞬間の該当するNRZI系列の状態を状態 β [k]として出力するビタビ復号器である。第二のビタビ復号器(VD)26は、ISIを差し引かれたあとの再生信号Y[k']に基づいて、1または0の2値の識別値B'[k'-m]を出力するPRIV用ビタビ復号器である。

【0083】以下、 β []で示される信号はPRIV該当a列の状態、mはビタビ復号器(VD)で識別結果が確定するまでの時間(動作周期)である。

【0084】アドレス発生器27は、第一のビタビ復号器25による仮識別値B[i] ($i = k-m-2n, \dots, k-m$ ただし、 $i \neq k-n$)の2nビット分の組合せと状態 β [k-m-n-1], 状態 β [k-m-n-2]を、(2n+2)ビットのアドレス{A}としてルックアップテーブル14に与えるアドレス発生回路である。(2n+m)段のディレーライン28は、再生信号X[k]を(m+n)動作周期分遅延させて遅延出力X[k']を出力する遅延回路である。ここで説明しない第四の再生等化器4の各部分は、第一の再生等化器1の同一符号を付した部分に相当する。また、図7中に点線で囲った部分を単にNLC10と呼ぶ。

【0085】以下、第四の再生等化器4の動作について説明する。第一のビタビ復号器25は、線型等化器(図示せず)の出力信号X[k]を仮識別し、1または0の2値の仮識別値B[k-m]と各瞬間の該当するNRZI系列の状態を状態 β [k]として出力する。ここでmはVDによって識別値が確定するまでのビット数であ

る。

【0086】アドレス発生器27は、第一のビタビ復号器25から出力される仮識別値B[i] ($i = k-m-2n, \dots, k-m$ ただし、 $i \neq k-n$)の2nビット分の組合せと状態 β [k-m-n-1], 状態 β [k-m-n-2]を、2n+2ビットのアドレス{A}をルックアップテーブル14に出力する。

【0087】このアドレス{A}を受けたルックアップテーブル14は、対応するISIデータを演算回路16に出力する。(2n+m)段のディレーライン28は再生信号X[k]を(m+n)動作周期分遅延させた遅延信号X[k']を演算回路16に入力する。ここで、($k' = k-n-m$)である。演算回路16は、遅延信号X[k']から前記ISIデータを減算し、第二のビタビ復号器26に出力する。第二のビタビ復号器26は、遅延信号X[k']から前記ISIデータを減算した再生信号Y[k']に基づいて1または0の2値の識別値B'[k'-m]を出力する。

【0088】ここで用いたビタビ復号器25、26とアドレス発生器27についてさらに説明する。第一の実施例～第四の実施例で述べた各NLCにおいては、PRIVに応用するためにはPRIVデコーダーが必要であった。しかし、ビタビ復号器25、26の出力はすでにデコードされており、1に識別されたときの符号は偶数系列または奇数系列のNRZIの状態としてビタビ復号器25、26の内部で復号に用いられている。

【0089】したがって、これをアドレスの決定に用いることで回路を簡略化することができる。ビタビ復号器25、26はNRZI符号用ビタビ復号器を偶数系列と奇数系列用に2並列に使用し、PRIV用のビタビ復号器としたものである。ここで状態 β という値が用いられているが、これが各系列の状態を表している。通常のビタビ復号器では仮識別値B[k-m]だけを出力し、状態 β は内部で用いられるだけであるが、本発明におけるビタビ復号器25、26では、kという瞬間の該当するNRZI系列の状態 β を状態 β [k]として出力している。

【0090】一方、ある瞬間kにアドレス{A}として必要なのは、仮識別値B[k-m-n]が確定する直前の各系列の状態である。したがって、アドレス発生器27は状態 β [k]をm+n-1動作周期分遅延させるm+n-1段のレジスターと、仮識別値B[k-m]を格納する2n段のレジスターによって構成され、状態 β [k-m-n-1], 状態 β [k-m-n-2]と仮識別値B[i] ($i = k-m-2n, \dots, k-m$ ただし、 $i \neq k-n$)を組合せてアドレス{A}とする。

【0091】以下、第六の実施例について説明する。図8は、本発明の第五の再生等化器5の構成を示す図である。第五の再生等化器5は、第四の再生等化器4に前記ISI計算回路18を加えたものである。図8において、ISI計算回路18は(2n+m)段のディレーラ

イン28の出力信号 $X[k']$ と第二のビタビ復号器26による識別値 $B'[k'-m]$ および状態 $\beta[k']$ から、内部のディレーによって $X[k'-n-m]$ 、状態 $\beta[k'-m-n-1]$ 、状態 $\beta[k'-m-n-2]$ 、識別値 $B'[i]$ ($i=k-m-2n, \dots, k-m$)を作り、これらを用いてISIデータを計算して、適時、ルックアップテーブルのISIデータを更新する計算回路である。その他の第五の再生等化器5の各部分は、第四の再生等化器4について同一の符号を付した各部分に同じである。

【0092】図8に示した第五の再生等化器5のような構成とすることにより、ISIデータの設定が不要なビット識別器を実現することができる。なお、この場合はISI計算回路18の内部でRAMのアドレスを計算する必要があるため、第二のビタビ復号器26からも状態 $\beta[k']$ を出している。ISI計算回路18は、以上に述べた各実施例で説明したISI計算回路18のいずれもが適用可能である。

【0093】以下、第七の実施例について説明する。図9は、本発明の第六の再生等化器6の構成を示す図である。第六の再生等化器6は、第二のビタビ復号器26、アドレス発生器27、 $(n+m)$ 段のディレーライン28、およびルックアップテーブル14、つまり、NLCと第一のビタビ復号器26を組み合わせ単位9とし、それを多段接続したものである。以下、 $X[k(j)]$ で示される信号は、 $(n+m)$ 段のディレーライン28をj回通った再生信号、 $B[k(N)]$ で示される信号は、N段の組み合わせ単位(NLC+VD)9による識別結果を示す。

【0094】この組み合わせ単位9の構成を図10に示す。図10に示す各部分は、第五の再生等化器5の同一符号を付した各部分に同じである。

【0095】第六の再生等化器6においては、線形等化後の再生信号を遅延させた $X[k-j(n+m)]$ とj段目の第二のビタビ復号器26による識別値 $B'[k-j(n+m)-m]$ を次の段に渡している。つまり、第六の再生等化器6は、図9に示すように上記組み合わせ単位を多段結合して前後に第一のビタビ復号器25とISI計算回路18を配する構成のPRIV用のビット識別器である。第六の再生等化器6では、より信頼性の高い仮識別値によってISIデータを読み出して再生信号から差し引くことで、第四の再生等化器4および第五の再生等化器5よりさらにエラーレートを向上させることができる。

【0096】また、ISIデータは最終段のVDの識別値に基づいて作成されるので、第五の再生等化器5よりもデータ自体の信頼性が高くなる。なお、エラーレートの上限は等化後の再生信号のS/Nによって規定されるので、前記組み合わせ単位の段数を多くとればいくらかでもエラーレートが向上するわけではなく、段数を多くし

ていくと $(B'[k-j(n+m)-m])=B'[k-(j+1)(n+m)-m])$ となって飽和する。したがって、2~3段の前記組み合わせ単位で十分である。

【0097】図11は、実験用デジタルVTRに511周期のM系列を記録し、等化前の再生信号を高速でAD変換して計算機に取り込んで、3~15タップのトランスバーサルフィルタでPRIVの基準に等化し、識別値と記録データを比較した場合のエラーレートを示す図である。

【0098】連続した再生信号データの数約2000個であり、この実験用デジタルVTRの1トラック分に相当する。NLCは $n=3$ のものを適用し、ISIデータの初期値はすべて0を入れて識別が進むにしたがって自動的に更新している。図11でTDと書いたのは閾値検出の結果であり、第五の再生等化器5に相当するビット識別器を適用した結果がVD→NLC→VDで示したものである。この中で最も大きなエラーレートの向上が得られているのは7タップのトランスバーサルフィルタに本発明を適用した場合であり、TDの(1/100)、単体のVDおよびNLCに比べて、約(1/10)にエラーが減少している。

【0099】

【発明の効果】以上述べたように本発明によれば、テールルックアップ型のNLCにおいて、初期設定も含めて外部からの調整を必要とせず、また、十分に長い前後のビットを考慮でき、また、テープ、ヘッドの特性の変化やばらつきを補償することができ、また、仮識別値に含まれるエラーを少なくすることができ、また、仮識別値にエラーが含まれていてもエラーの伝播が少なく、また、符号間干渉が減少するとともにノイズの相関が減少し、信号電力有効利用を図ることができ、また、簡単な回路構成で実現でき、良好なエラーレートが得られるノンリニアキャンセラーを提供することができる。

【図面の簡単な説明】

【図1】本発明の第一の再生等化器の構成を示す図である。

【図2】ISI計算回路の構成を示す図である。

【図3】本発明の第二の再生等化器の構成を示す図である。

【図4】本発明の第三の再生等化器の構成を示す図である。

【図5】本発明の第三の再生等化器において、 $n=3$ の場合を例に、仮識別値 $B[k]$ 、仮識別値 $A[k]$ および $(2n+2)$ ビットのアドレス{A}の関係の例を示す図である。

【図6】実験用デジタルVTRに511周期のM系列を記録し、等化前の再生信号を高速でAD変換して計算機に取り込んで、3~15タップのトランスバーサルフィルタでPRIVの基準に等化し、識別値を記録データと比較したときのエラーレートを示す図である。

【図7】本発明の第四の再生等化器の構成を示す図である。

【図8】本発明の第五の再生等化器の構成を示す図である。

【図9】本発明の第六の再生等化器の構成を示す図である。

【図10】第六の再生等化器の組み合わせ単位の構成を示す図である。

【図11】実験用デジタルVTRに511周期のM系列を記録し、等化前の再生信号を高速でAD変換して計算機に取り込んで、3～15タップのトランスバーサルフィルタでPRIVの基準に等化し、識別値と記録データを比較した場合のエラーレートを示す図である。

【図12】従来の再生等化器の構成を示す図である。

【符号の説明】

- 1・・・第一の再生等化器
- 2・・・第二の再生等化器
- 3・・・第三の再生等化器

4・・・第四の再生等化器

5・・・第五の再生等化器

6・・・第六の再生等化器

9・・・組み合わせ単位

10・・・NLC

11・・・線型等化器

12・・・第一の2値識別器

13・・・2n段のディレーライン

14・・・ルックアップテーブル

15・・・n段のディレーライン

16・・・演算回路

17・・・第二の2値識別器

18・・・ISI計算回路

21、22・・・3値識別器

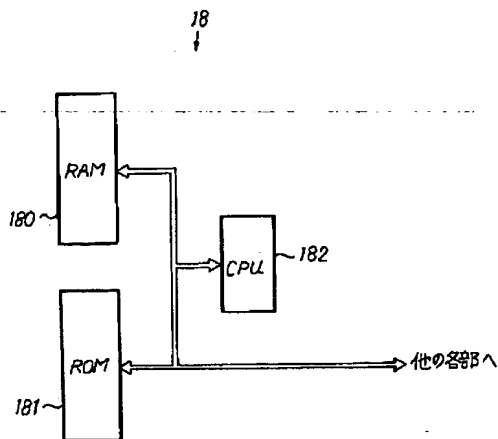
23、24・・・PRIVデコーダー

25、26・・・ビタビ復号器

27・・・アドレス発生器

28・・・(2n+m)段のディレーライン

【図2】



【図5】

i	K6	K5	K4	K3	K2	K1	K	Add	Aeven
B(i)	-1	1	0	-1	0	0	1	*	*
A(i)	1	1	0	1	0	0	1	*	*
[A]	1	1	0	*	0	0	1	0	1

B(K): ノンリニア・キャンセラ
識別値(1,0,-1)

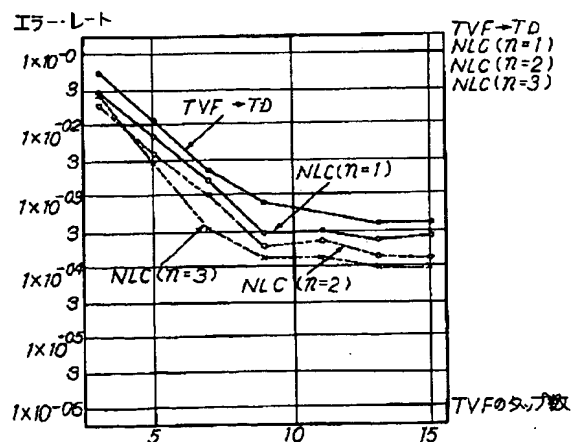
A(K): PRIVデコード後の
ノンリニア・キャンセラ
識別値(1or0)

i	K6	K5	K4	K3	K2	K1	K	Add	Aeven
B(i)	1	-1	0	1	0	0	-1	*	*
A(i)	1	1	0	1	0	0	1	*	*
[A]	1	1	0	*	0	0	1	1	0

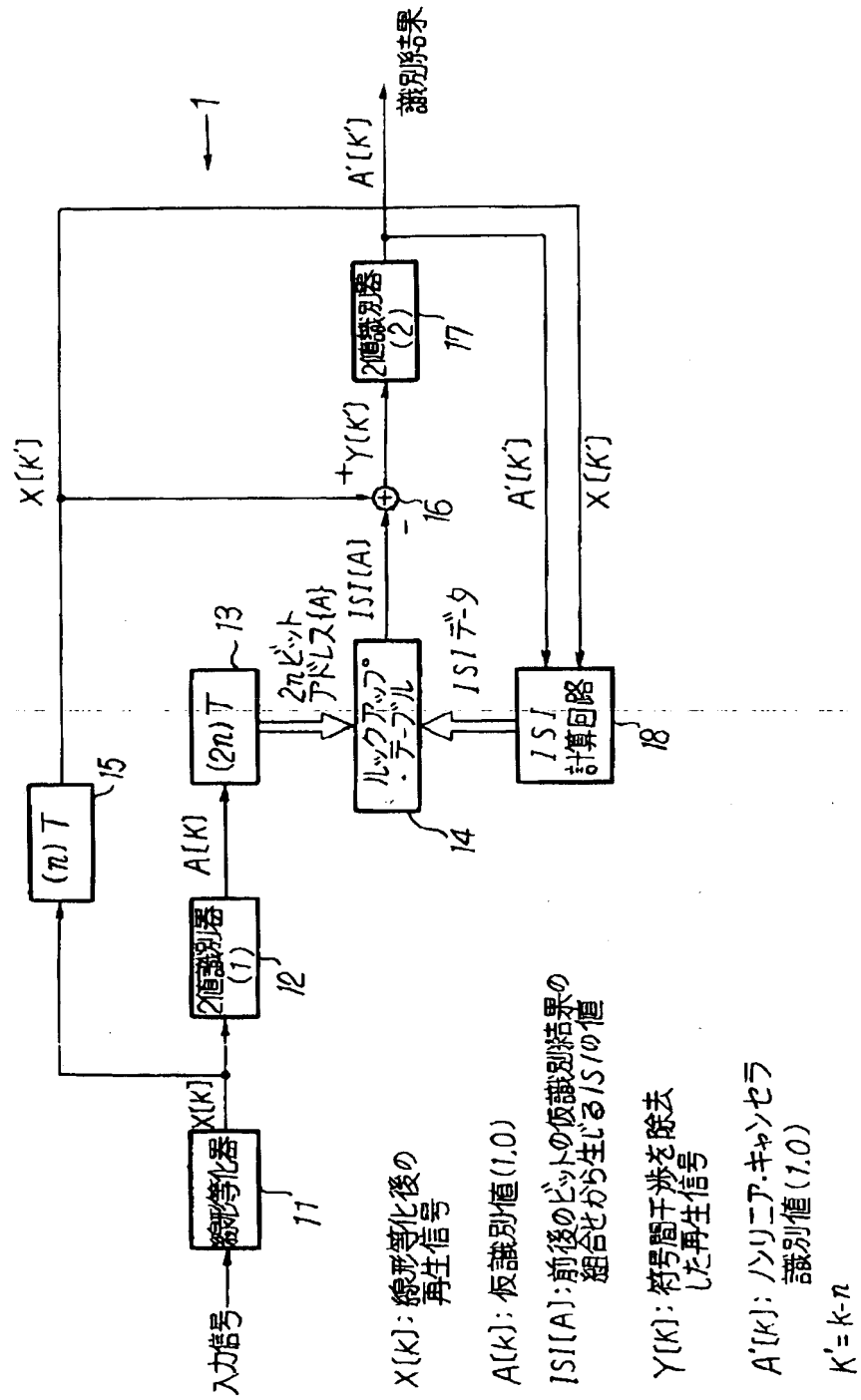
Add: 奇数個前のNRZI系列
の状態(1or0)

Aeven: 偶数個前のNRZI系列
の状態(1or0)

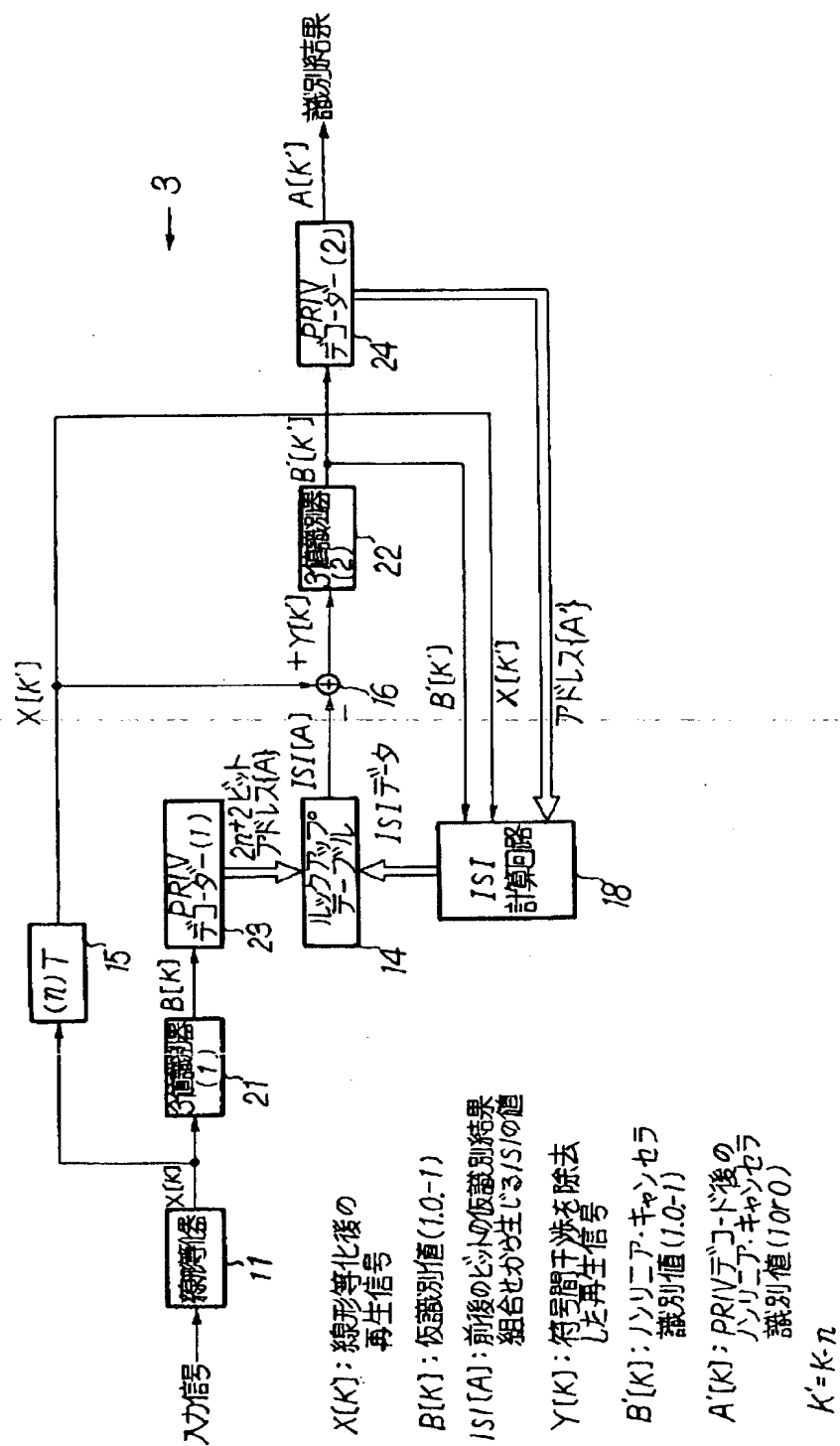
【図6】



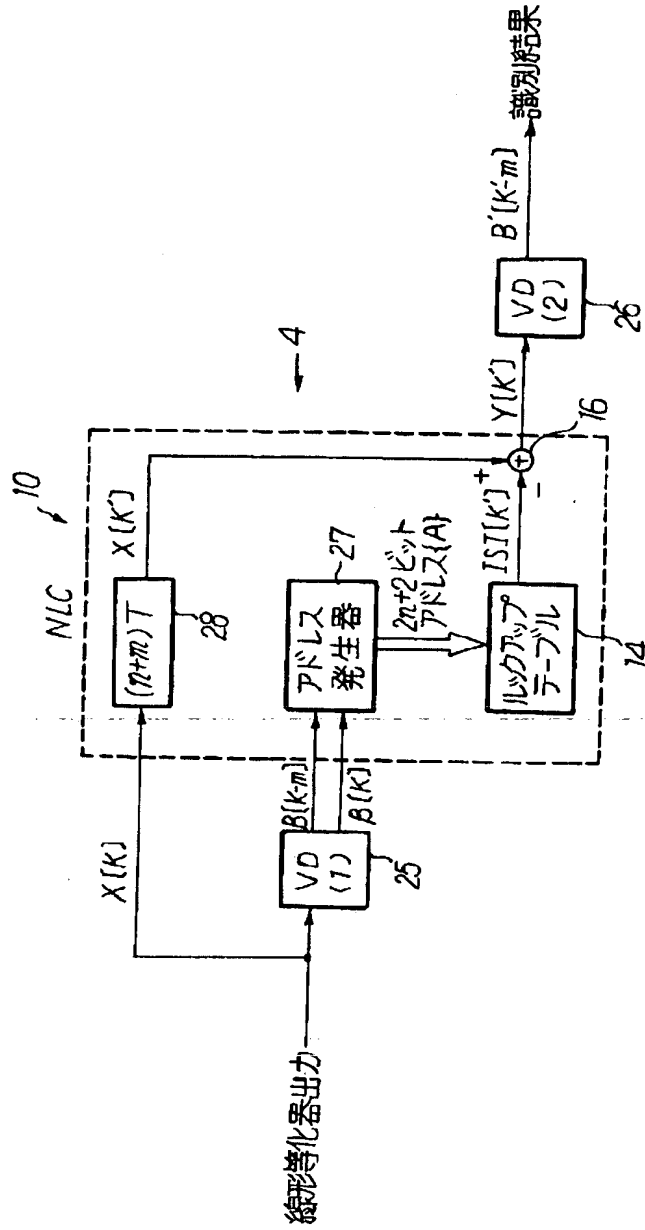
【図1】



【図4】



【図7】



$X(k)$: 線形等化後の再生信号

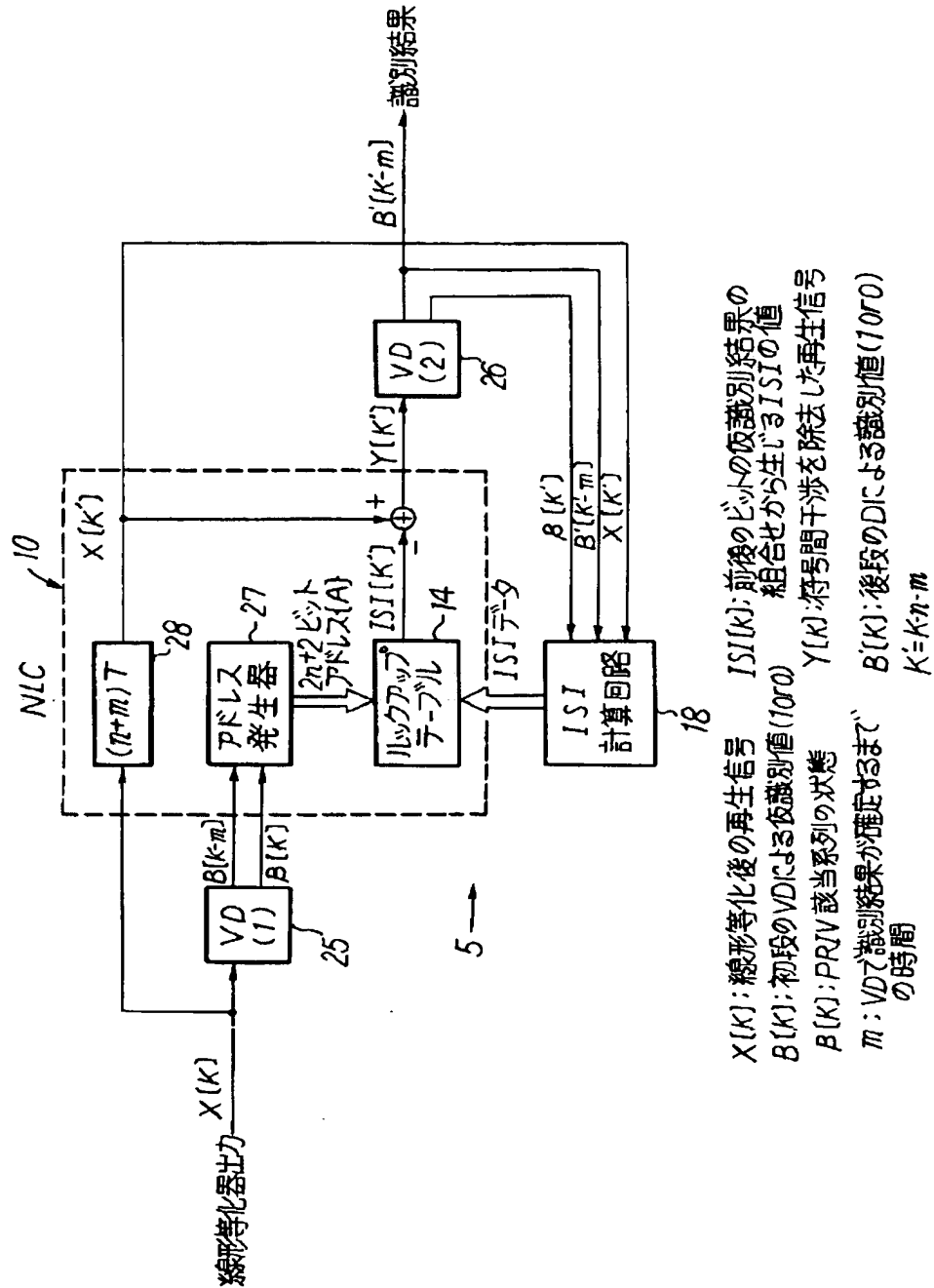
$B(k)$: 初段のVDによる仮識別値(1or0) $ISI(k)$: 前後のビットの仮識別の組合せから生じるISIの値

$B(k)$: PRIV該当系列の状態 $Y(k)$: 符号間干渉を除去した再生信号

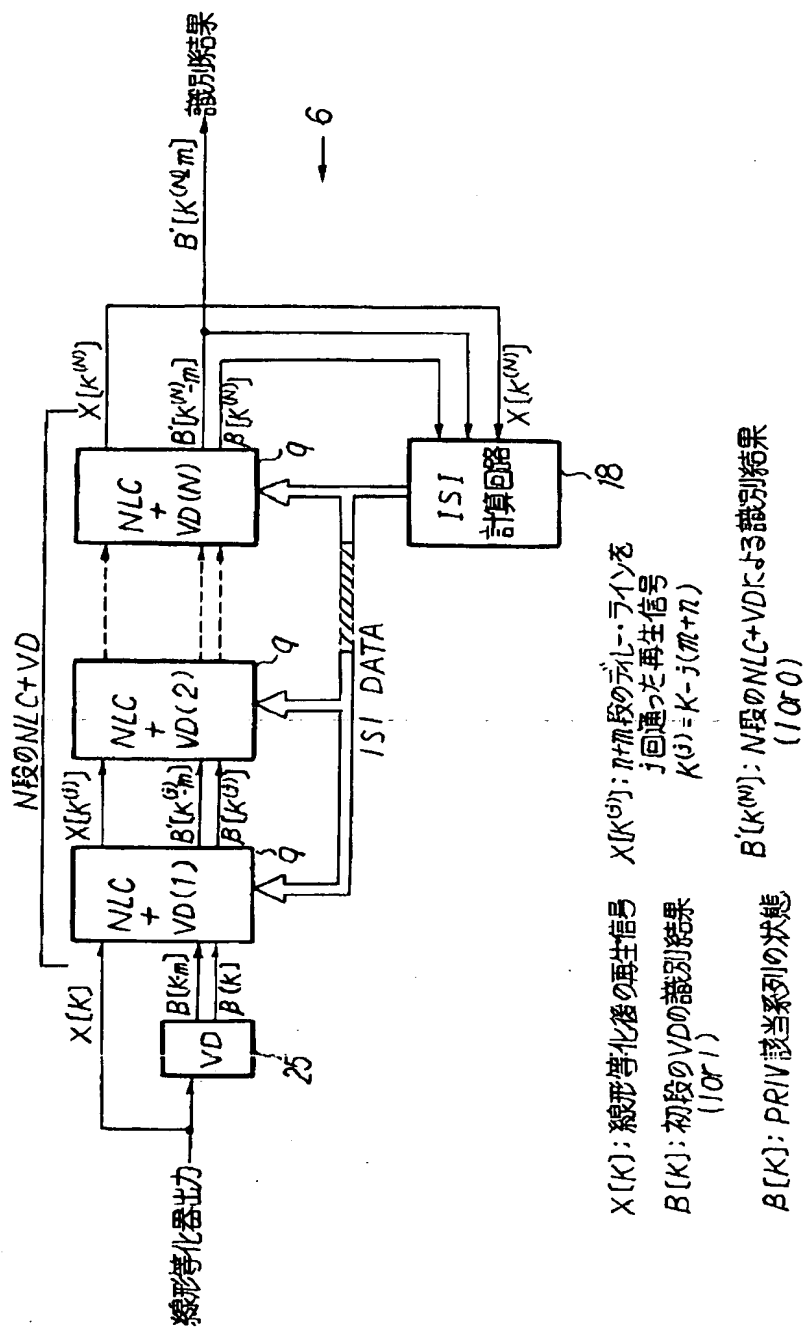
m : VDで識別結果が確定するまでの時間 $B'(k)$: 後段のVDによる識別値(1or0)

$k' = k - n - m$

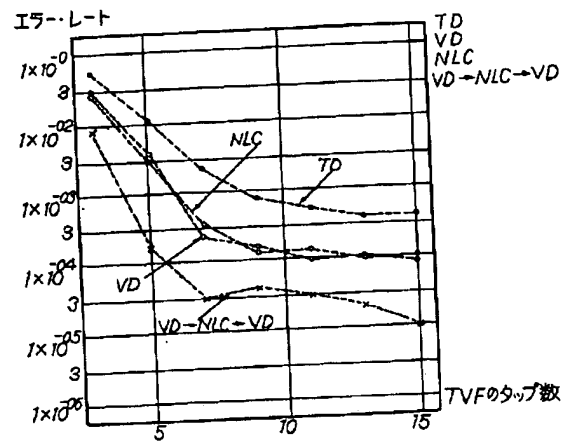
【図8】



【図9】



【図11】



フロントページの続き

(51) Int. Cl. 5

H 04 B 3/14

H 04 N 5/92

識別記号

庁内整理番号

F I

技術表示箇所

8226-5K

H 4227-5C